Diese Website verwendet Cookies von Google, um Dienste anzubieten und den Traffic zu analysieren. Deine IP-Adresse und dein User-Agent werden zusammen mit Messwerten zur Leistung und Sicherheit für Google freigegeben. So können Nutzungsstatistiken generiert, Missbrauchsfälle erkannt und behoben und die Qualität des Dienstes gewährleistet werden.

WEITERE INFORMATIONEN OK

Friday, August 24, 2007

Blind signatures

The following, from a larger article of mine, serves as an introduction to the idea of blind signatures and their use in digital cash and other digital bearer certificates:

Introduction

Meet the greatest simple equation since $e=mc^2$:

$$gSf(m) = S(m)$$

S is a digital signature, f is the blinding function, and g an unblinding function. The blinding functions are usually based on a secret random number called the "blinding factor". m is another random number, a unique identifier which can, for example, refer to an instance of some object.

The idea is very clever but very simple. It may be counterintuitive because the simplest physical world metaphor of this highly useful e-commerce primitive sounds worse than useless: Alice can get Carol to sign a blank check! Here's how:

(1) Alice generates m and blinds it. "Blinding" is just a one-time-pad encryption to oneself, f(m). She sends this to Carol. This is like taking a piece of paper and sealing it inside an envelope which Carol can't open or see through.

(2) Carol signs it: Sf(m), and sends this back to Alice. This is like Carol signing the outside of the envelope.

(3) Alice unblinds it: gSf(m) = S(m). Carol has also signed the paper Alice put inside the envelope!

The genius behind this discovery: cryptography guru David Chaum. The brilliance lies in step 3: Chaum discovered that some signatures have the property of being "commutative" with the blinding functions: Alice can strip off the blinding in the reverse order which the blinding and signature were applied, leaving just Alice's signature of n. It is as if Alice put a piece of carbon paper inside the envelope.

Pages

- Home

About Me

Nick Szabo

"A premier thinker about history, law and economics, and the lessons they have for security." -- Adam Shostack, Emergent Chaos

"Szabo comes out with these essays that leave me in awe." -- Brian Dunbar

"Reading material that is eclectic, challenging and endlessly fascinating." -- Sean McGrath, Propylon

"Like most blogs worth my attention, this blog is updated only infrequently. That is because the authors of blogs worth my attention only post when they have something to say that is true, relevant and not already known by their audience. Most of the human race does not have the skill to know when an idea has these three properties. The skill is particularly rare in the fields of politics and economics, which is why this blog is such a rare and valuable thing." -- Richard Hollerith

View my complete profile

In particular for RSA signatures, with public key (pq, e) and private key d, the blind signature functions are the following modulo pq:

$$S(x) = x^d$$
$$g(x) = xk^{-1}$$
$$f(x) = xk^e$$

We can check that the blind signature property holds:

$$gSf(m) = (m(k^e))^d * k^{-1}$$
$$= m^d * k * k^{-1}$$
$$= m^d$$

which is the valid RSA signature of private key d on m.

Unlinkable Transfers

Distinguish between either a counter or third party tracing one person's true name, via lack of or weak communications mix, and a third party linking two entities (whether nyms, use-more-than-once-addresses, account numbers, or true names) as being involved in the same transaction. By unlinkability herein we mean the latter. The goal where true names are used (this occurs, for example, when using true name accounts or not using good communications mixes), is to prevent third party linking of two people doing business with each other. Where nyms are used the goal is to minimize the release of traffic information, to prevent the unwanted accumulation of unique behavior patterns, which could be used to link nyms (including to their true names), or could augment other means of breaching privacy. Blinding especially helps where rights holders want to keep third party or public accounts denominated in generic rights. In that case a communications mix doesn't even in principle give us what blinding does.

Besides protecting against the transfer agent, Chaum's transferor-, transferee-, and double-blinding protocols protect against collusion of a party with a transfer agent to identify the countparty account or nym.

Unlinkability can be provided by combining a list of cleared certificates with blind signatures and a delay-mixing effect. Enough instances of a standardized contract [or specifically with digital cash, standard denominations of money] are issued over a period of time to create a mix. Between the issuing and clearing of a certificate, many other certificates with the same signature will be cleared, making it highly improbable that a particular clearing can be linked to a particular issue via the signature. There is a tradeoff between the mixing effect and the exposure to the theft of a "plate" for a particular issue: the smaller the issue, the smaller the exposure but the greater the linkability; a larger issue has both greater exposure and greater confidentiality.

Blind signatures can be used to make certificate transfers unlinkable via serial number. Privacy from the transfer agent can take the form of transferee- unlinkability, transferor-unlinkability, or "double blinded" where both transferor and transferee are unlinkable by the transfer agent or a collusion of a transfer agent and counterparty.

A use-once-address communications mix plus foreswearing any reputation gain from keeping accounts, in theory also buys us unlinkability, but a communications mix is weak and very expensive.

Bearer certificates come in an "online" variety, cleared during every transfer, and thus both verifiable and observable, and an "offline" variety, which can be transferred without being cleared, but is only verifiable when finally cleared, by

Blog Archive

- ► 2018 (1)
- ► 2017 (3)
- ► 2016 (4)
- ► 2015 (3)
- ► 2014 (3)
- ► 2013 (3)
- ► 2012 (8)
- ► 2011 (12)
- ► 2010 (9)
- ► 2009 (29)
- ► 2008 (55)
- ▼ 2007 (47)
  - ► December (1)
  - ► October (2)
  - ► September (8)
  - ▼ August (10)
    - Money and the problem of cooperation
    - Why the industrial revolution?
    - Mutually private computation
    - Blind signatures
    - Personal jurisdiction before sovereignty
    - Smart contracts watch
    - Exit and freedom
    - Our new omniscient savior
    - Ten ways to make a political difference
    - Government for profit
  - ► July (4)
  - ► June (4)
  - ► May (8)
  - ► April (1)
  - ► March (2)
  - ► February (4)
  - ► January (3)

revealing any the clearing name of any intermediate holder who transferred the object multiple times (a breach of contract).

This unlinkability is often called "anonymity", but the issue of whether accounts are issued to real names or pseudonyms, and whether transferor and transferee identify themselves to each other, is orthogonal to unlinkability by the transfer agent in the online model. In the off-line model, account identification (or at least a highly reputable and/or secured pseudonym) is required: passing an offline certificate a second time reveals this identity. Furthermore, communications channels can allow Eve to link transferor and transferee, unless they take the precaution of using an anonymous remailer. Online clearing does make lack of identification a reasonable option for many kinds of transactions, although common credit and warrantee situations often benefit from or even require identification.

When confronting an attempted clearing of a cleared serial number, we face an error-or-fraud dilemma similar to the one we encountered above in double entry bookkeeping. The ecash(tm) protocol from DigiCash actually takes advantage of on purpose to recover from a network failure. When certificates are lost over the net it is not clear to the transferor whether they have been received and cleared by the transferee or not. Second-transferring directly with the transfer agent resolves the ambiguity. This only works with the online protocol. The issue of distinguishing error from fraud is urgent in the offline protocol, but there is as yet no highly satisfactory solution. This problem is often intractable due to the subjectivity of intent.

With ideal two-way anonymous communications between use-once keys, and completely accountless clearing, unlinkability via blind signatures becomes redundant. This ideal case has yet to be even closely approached with implemented technology, and necessarily involves long communications delays which are often intolerable. Real imperfect communications mixes and less expensive blinded tokens complement each other.

Posted by Nick Szabo at 1:39 PM

► 2006 (130)
► 2005 (44)

12 comments:

Daniel A. Nagy said...

In my teaching practice, I found the following metaphors useful:

Digital signatures are actually closer to seals, in that the private key, just like the stamp, can be lost or stolen.

Blinded signatures are like embossing: you put your piece of paper into an envelope, let the notary emboss it with her stamp and then retrieve it from the envelope.

However, there is a crucial difference between paper-based and digital signatures, which has very far-reaching implications for carrying over commercial law to cyberspace:

Once a piece of paper has been signed, the unsigned version ceases to exist. In particular, once you endorse a cheque, it stays endorsed. Not true with digital signatures! Digital signatures can be removed without trace.

This results in the general problem of double spending. It is not difficult to detect double spending *post factum* (in the above example, whoever forks the chain of endorsements on a negotiable instrument is a fraud), but it is far more difficult to prevent it. *A*

posteriori detection, while much better than nothing, is a poor substitute for proactive measures. In particular, the whole point of negotiable instruments is preventing insolvency by letting the recipient worry about only three things: the creditworthiness of the original issuer, the honesty of the last endorser and the anti-forgery measures of the instrument. If these are in order then there is no reason not to accept that piece of paper. Doing the same digitally is quite a challenge, because one can never know whether a fork occurred, until it surfaces, which might be too late.

Thus, we need some irreversible operation in the digital world, similar to marking a piece of paper with ink. One such operation is making a secret public. It can be done, but cannot be reversed. Public information cannot be made secret again (or, more precisely, the costs of doing so can be easily made prohibitive). This is why I expect time-stamping services publishing hashes of signed documents to play a central role in the future of cyberlaw.

2:50 PM

Anonymous said...

The number used in the blinding funtion is usually random, but suppose Alice wishes to get multiple signatures for the price of one....

Alice commutatively merges N many documents and after the signature is applied removes different sets of N-1 of them to recover N signed docs.

What stops her doing this?

3:25 PM

nick said...

DN: "It is not difficult to detect double spending post factum (in the above example, whoever forks the chain of endorsements on a negotiable instrument is a fraud), but it is far more difficult to prevent it. "

I assume you are referring just to offline transfers? With online clearing preventing double-spending is trivial -- the clearer keeps a spent numbers list.

3:40 PM

Anonymous said...

if you allow a party to hold on to their token for a significant amount of time (meaning the mixing time is long, and you're approaching bearer certificates), rather than merely using this to create a temporary mix where everybody agrees to play musical chairs at a certain time, users have no way of knowing how many tokens have actually been issued (by the bank inflating the tokens, or the key being stolen). engineering cannot solve this problem, which certainly wouldn't inspire confidence in the system.

11:56 AM

nick said...

anonymous: "users have no way of knowing how many tokens have actually been issued (by the bank inflating the tokens, or the key being stolen)."

This is quite true for digital cash as described in the cryptographic literature, and as it has been implemented in ecash and lucre, and AFAIK as it is still currently implemented. The issuer and clearer (in my terminology a "mint" or "bank" is divided up into two potentially separate entities, the issuer and the clearer) are trusted third parties, which means that they are security holes that must be closed by means outside the digital cash protocol itself.

For digital cash or other blinded bearer certificates to be ecnomically viable users have to get value out of the system, in terms of satisfying their preferences for private and irreversible transactions, faster than the rate inflation. There are, short of the stronger methods addressed below, but just within the single-issuer/single clearer paradigm, ways for third parties to audit the issuer and clearer and track the rate of inflation simply by posing as customers or gathering sampling information from customers. This reputation system puts a practical limit on the expected rate (or risk per unit time) of inflation but is admittedly an inferior solution to those named below. Also digital cash could piggyback on the reputation and gold redemption windows of the gold currency issuers -- it would thereby be protected against inflation to the same extent that the current insecure (in terms of privacy and reversibility) gold currencies are protected.

anonymous: "engineering cannot solve this problem"

That's a preposterously strong claim given that you don't provide any evidence for it, much less the proof you would need to actually demonstrate it. In fact there are a number of candidates solutions. Google the following: "quorum systems" (or just Byzantine replication generally), "secure property titles", "remote attestation", "bit gold".

Bit gold, for example, can be tied to digital cash through a redemption window, as per the old free banking note issue and redemption system. As long as the window keeps working users can have substantial confidence that the system is working -- the temporal window you pointed out becomes much longer.

Remote attestation and Byzantine replication are alternative and complementary ways to distribute issuers and clearers and force them to run publically verifiable code, so that a single clearer or even a small conspiracy of clearers can't undetectably inflate the currency.

2:50 PM

nick said...

anonymous: "Alice commutatively merges N many documents and after the signature is applied removes different sets of N-1 of them to recover N signed docs."

I don't personally know the answer to this -- perhaps a fellow reader who has kept closer track of the cryptogrpahic literature than I can chime in. I am confident that almost any of the dozens of cryptography scholars, and almost surely Chaum himself, who have tried to break digital cash, anonymous voting, blinded credentials, and other protocols based on blind signatures that have been developed over the last 20 years have thought of this kind of attack, and if it is actually a problem have figured out solution(s) to it.

BTW, I don't recommend you go out an implement digital cash simply based on my description. For one thing, there are already open source public-domain implementations out there done by experts in the field such as Ben Laurie. You should either reuse them or at least learn from them before doing your own. For another, my description is meant to be an introductory/beginner demonstration, not a complete blueprint for implementing secure blinding. The purpose of the article is to alert people to the existence of this protocol and give them some of the basic information they need to understand the literature or to understand the basics of how blind signatures work if they are looking for a privacy solution. The next step is then to either use an implementation done by an expert in the field (for example, Ben Laurie's lucre) or to become knowledgeable yourself, if you are mathematically minded, by reading some of the copious cryptographic literature on blinding. You should not go implement the sketch I have given here and call it secure based on my authority.

3:06 PM

Anonymous said...

Well I think I've figured out the position with Alice trying to get multiple signatures for the price of one.

Alice CAN get multiple signatures from each one provided by Bob, but they won't be interesting because she won't have enough control over what they are.

It's the same position that Alice is in by knowing Bob's public key - she can encrypt a blob M to produce C and then switch the C and M blobs and tell everybody "Bob signed this number here" - but because it's meaningless (and not an instruction or statement worth signing) it is easily dismissed.

At least that's the way it seems in Schneier's section 23.12. Section 5.3 which shows less detail makes the attack look workable.

2:50 PM

nick said...

Anonymous: that's right, and in fact the list of long random numbers to represent issued certificates, kept on a spent list used to clear certificate transfers, means that Alice can't guess the number of a certificate to forge it.

3:53 PM

Anonymous said...

Can blind signatures be implemented with hashing? Alice hashes message m with a salt and have Bob sign it. Later to verify, Alice reveals the salt and the message. Does this work? It there any reason to use more complicated math?

11:12 PM

Anonymous said...

good post

3:40 AM

Anonymous said...

Thanks

2:50 AM

Anonymous said...

Ive been doing research around cryptonote technology and found this post. I dont know if these two are linked. I am kinda a cryptography noob.

Maybe its of some interest to someone here: https://en.wikipedia.org/wiki/CryptoNote

12:10 PM

Post a Comment

Simple theme. Powered by Blogger.